

**AFRL-SN-WP-TM-2003-1031**

**CONTEXT BASED TEMPORAL  
REASONING AND ANALYSIS OF  
VEHICULAR ACTIVITIES FROM SAR  
EO IMAGERY**



**Benjamin B. Bederson**

**University of Maryland  
3171 A.V. Williams Bldg.  
College Park, MD 20742**

**JANUARY 2003**

**Final Report for 03 March 1997 – 16 January 2003**

**Approved for public release; distribution is unlimited.**


**SENSORS DIRECTORATE  
AIR FORCE RESEARCH LABORATORY  
AIR FORCE MATERIEL COMMAND  
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7318**

## NOTICE


Using government drawings, specifications, or other data included in this document for any purpose other than government procurement does not in any way obligate the U.S. Government. The fact that the government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey and rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report has been reviewed by the Office of Public Affairs (ASC/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.


This technical report has been reviewed and is approved for publication.



JASON J. JOHNSON, Project Engineer  
Find, Fix, Track and ID Branch  
Sensor ATR Technology Division  
Sensors Directorate



JOHN C. OSTGAARD, Chief  
Find, Fix, Track and ID Branch  
Sensor ATR Technology Division  
Sensors Directorate



EDMUND ZELNIO, Chief  
Sensor ATR Technology Division  
Sensors Directorate

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YY) January 2003		2. REPORT TYPE Final		3. DATES COVERED (From - To) 03/03/97 – 01/16/03	
4. TITLE AND SUBTITLE CONTEXT BASED TEMPORAL REASONING AND ANALYSIS OF VEHICULAR ACTIVITIES FROM SAR EO IMAGERY				5a. CONTRACT NUMBER F33615-97-1-1018	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 62301E	
6. AUTHOR(S) Benjamin B. Bederson				5d. PROJECT NUMBER ARPA	
				5e. TASK NUMBER AA	
				5f. WORK UNIT NUMBER 1P	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Maryland 3171 A.V. Williams Bldg. College Park, MD 20742				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Sensors Directorate Air Force Research Laboratory Air Force Materiel Command Wright-Patterson Air Force Base, OH 45433-7318				10. SPONSORING/MONITORING AGENCY ACRONYM(S) AFRL/SNAR	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-SN-WP-TM-2003-1031	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES See other published work in the DTIC collection under contract number F33615-97-1-1018. This report contains color.					
14. ABSTRACT (Maximum 200 Words) The objective of this effort was to extend the state of the art in Zoomable User Interfaces (ZUIs) and closely related interface techniques to support the Command Post of the Future (CPoF) and Semantic Web DARPA projects. This project had several important outcomes in the following areas, each of which will be described in more detail. In addition, all relevant published papers that were written as a result of this work are also included along with the amount of use of our work by others indicated by web access. In addition, a description of all invited presentations of this work is listed. The areas of work are: Jazz toolkit to support ZUIs, Piccolo toolkit to support ZUIs, PhotoMesa image browser, CounterPoint presentation tool, OZONE ontology navigator and Fisheye menus for selection within long lists.					
15. SUBJECT TERMS Zoomable User Interfaces (ZUI), Command Post of the Future (CPoF)					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT: SAR	18. NUMBER OF PAGES 16	19a. NAME OF RESPONSIBLE PERSON (Monitor) Jason Johnson 19b. TELEPHONE NUMBER (Include Area Code) (937) 255-5668 x4047
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			

## **Final Report**

Contract #F33615-97-1-1018

“Context based temporal reasoning and analysis of vehicular activities from SAR EO imagery”

January 16, 2003

Benjamin B. Bederson

[bederson@cs.umd.edu](mailto:bederson@cs.umd.edu)

The objective of this effort was to extend the state of the art in Zoomable User Interfaces (ZUIs) and closely related interface techniques to support the Command Post of the Future (CPOF) and Semantic Web DARPA projects.

This project had several important outcomes in the following areas, each of which will be described in more detail. In addition, all relevant published papers that were written as a result of this work are also included along with the amount of use of our work by others as indicated by web access. In addition, a description of all invited presentations of this work are listed. The areas of work are:

- Jazz toolkit to support ZUIs
- Piccolo toolkit to support ZUIs
- PhotoMesa image browser
- CounterPoint presentation tool
- OZONE ontology navigator
- Fisheye menus for selection within long lists

### **Jazz Toolkit**

Application developers rely on User Interface (UI) toolkits such as Microsoft's MFC and .NET Windows Forms, and Sun's Swing and AWT to create visual user interfaces. However, while these toolkits are effective for traditional forms-based applications, they fall short when the developer needs to build a new kind of user interface component – one that is not bundled with the toolkit. These components might be simple widgets, such as a range slider, or more complex objects, including interactive graphs and charts, sophisticated data displays, timeline editors, zoomable user interfaces, or fisheye visualizations.

Developing application-specific components usually requires large amounts of custom code to manage a range of features, many of which are similar from one component to the next. These include managing which areas of the window need repainting (called *region management*), repainting those regions efficiently, sending events to the internal object that is under the mouse pointer, managing multiple views, and integrating with the underlying windowing system.

Writing this code is cumbersome, yet most standard 2D UI toolkits provide only rudimentary support for creating custom components – typically just a set of methods for drawing 2D shapes, and methods for listening to low-level events.

Jazz is a new general-purpose toolkit for creating ZUI applications using zooming object-oriented 2D graphics. Jazz is built entirely in Java and runs on all platforms that support Java 2.

Jazz uses the Java2D renderer, and is organized to support efficient animation, rapid screen updates, and high quality stills. While we could have written Jazz using other rendering engines, such as OpenGL, we picked Java2D because of its clean design and focus on high-quality 2D graphics. As previously mentioned, OpenGL does not support business graphics well. In addition, using Java2D allows us to support embedded Swing widgets, which would be impossible with OpenGL.

Jazz borrows many of the structural elements common to 3D scene graph systems, such as Sun's Java3D [1] and SGI's OpenInventor [5]. By using a basic hierarchical scene graph model with cameras, Jazz is able to directly support a variety of common as well as forward-looking interface mechanisms. This includes hierarchical groups of objects with affine transforms (translation, scale, rotation and shear), layers, zooming, internal cameras (portals), lenses, semantic zooming, and multiple representations.

We and others have used Jazz extensively in a range of projects – with the ones relevant to this contract described below.

<http://www.cs.umd.edu/hcil/jazz>

- Unique accessors total: **135,700**
- Unique accessors weekly average: **750**
- Unique downloads total: **10,300**
- Unique downloads weekly average: **60**

## Piccolo Toolkit

Piccolo is a Java toolkit based on Jazz. We are also currently porting Piccolo to C#. It supports essentially the same core feature set (except for embedded Swing widgets), but its design is monolithic rather than polyolithic. This design change came from our experience building applications with Jazz. We found that the polyolithic approach in Jazz met our original design goals of being easy to understand, maintain and extend. But, managing all of the node types was too big a burden for the *application programmer*.

We call the concrete design approach adopted by most 2D toolkits *monolithic*, because these toolkits have a few large classes containing all the core functionality likely to be used by applications. We call the Piccolo-toolkit design approach *polyolithic*, because it consists of many small classes, each representing an isolated bit of functionality where several are often linked together to represent one semantic unit.

Monolithic toolkits suffer from a common problem: the toolkit classes tend to be complex and have large numbers of methods, and the functionality provided by each class is hard to reuse in new widgets. To support code reuse, toolkit designers often place

large amounts of generally useful code in the top-level class that is inherited by all of the widgets in the toolkit. This decision leads to a complex hard-to-learn top-level class. In addition, application developers are forced to accept the functionality provided by the toolkit's top-level class – they often cannot add their own reusable mechanisms to enhance the toolkit.

Polyolithic designs on the other hand, can potentially offer both reusability and customizability, because they compose functionality through runtime instantiation rather than through sub-classing. This promise of better toolkit maintainability and extensibility led us to the polyolithic design of Jazz.

Piccolo gives up on the idea of separating each feature into a different class, and instead puts all the core functionality into the base object class, `PNode`. Piccolo also eliminates the separation between “node” and “visual component” types. Instead, every node can have a visual characteristic. In practice, this nearly halves the number of objects since most nodes ended up having a visual representation in Jazz, requiring two objects.

The Piccolo `PNode` class is thus bigger than Jazz's `ZNode` class, having 140 public methods compared with Jazz's 64 public methods. Piccolo also uses a scene graph and supports hierarchies, transforms, layers, zooming, internal cameras, etc. as does Jazz. The “Hello World” program in Piccolo looks very similar to the Jazz version.

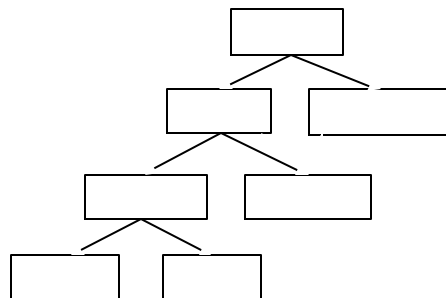
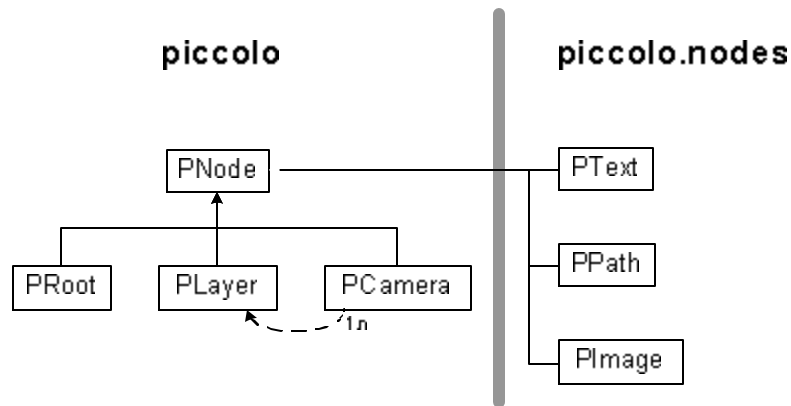
```
import edu.umd.cs.piccolo.nodes.*;
import edu.umd.cs.piccolox.*;

public class PHelloWorld extends PFrame {
    public void initialize() {
        PText text = new PText("Hello World!");
        getCanvas().getLayer().addChild(text);
    }

    public static void main(String args[]) {
        new PHelloWorld();
    }
}
```

**Piccolo “Hello World!” program that supports panning and zooming. Or, one can create a “PCanvas” and place that anywhere a Swing `JComponent` can go.**

The Piccolo object hierarchy is also similar to Jazz, but again, is greatly simplified since many node types are merged into the core class. There are also fewer visual node types because they have been generalized.



<http://www.cs.umd.edu/hcil/piccolo>

- Unique downloads total: 1,050
- Unique downloads weekly average: 45

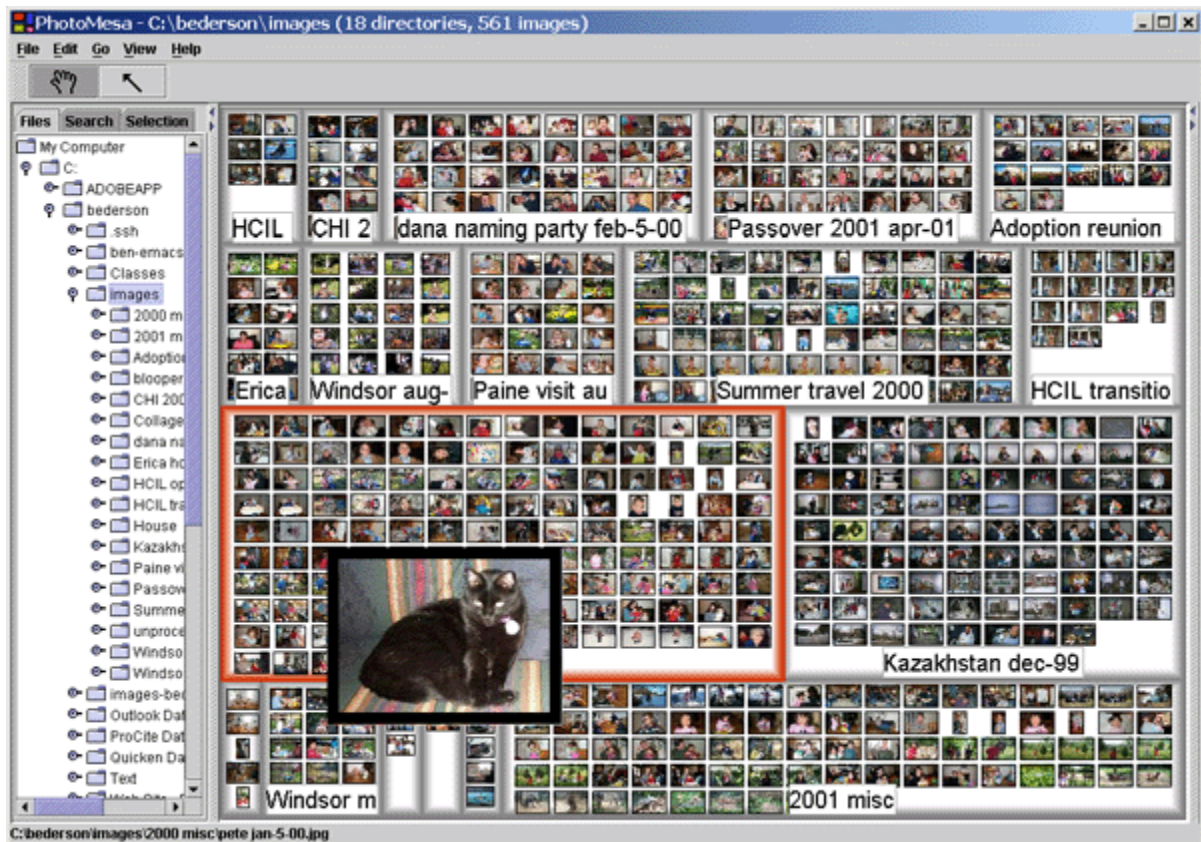
## PhotoMesa Image Browser

PhotoMesa is a zoomable image browser that uses a novel treemap algorithm to present large numbers of images grouped by directory, or other available metadata. It uses a new interaction technique for zoomable user interfaces designed for novices and family use that makes it straightforward to navigate through the space of images, and impossible to get lost.

PhotoMesa groups images using one of two new algorithms that lay out groups of objects in a 2D space-filling manner. *Quantum treemaps* are designed for laying out images or

other objects of indivisible (quantum) size. They are a variation on existing treemap algorithms in that they guarantee that every generated rectangle will have a width and height that are an integral multiple of an input object size. *Bubblemaps* also fill space with groups of quantum-sized objects, but generate non-rectangular blobs, and utilize space more efficiently.

Image browsing is important for a number of reasons. First of all, no matter what information retrieval system is being used, the user has to browse the results of the search. It is certainly important to build query systems that help users get results that are as close to what is wanted as possible. But there will always be images that need to be browsed visually to make the final pick.



**PhotoMesa: A Zoomable Image Browser**

Most image browsing systems present the images as a grid of thumbnails that the user can scroll through with a vertical scrollbar, and see a high resolution version of the image with some mouse interaction. There are also a few alternative designs, such as manually constructed digital photo albums, and one commercial zoomable image browser.

A second reason for needing new image browsers is more subtle, and was actually my primary motivation for doing the present work. Sometimes, people browse images just for the pleasure of looking at those images, and they often do it with other people. This is especially true for personal photos. As people take more digital family pictures, we need better tools to support users in home settings as they look at those pictures together on a computer screen. Looking at home photos has a lot of overlap with traditional retrieval



systems. People still want to be able to find photos of particular people and events, etc. However, they are less likely to be time pressured to find a particular photo, and more likely to be interested in serendipity – that is, finding photos they weren't looking for.

<http://www.cs.umd.edu/hcil/photomesa>

- Unique accessors total: **75,400**
- Unique accessors weekly average: **875**
- Unique downloads total: **31,700**
- Unique downloads weekly average: **370**
- Total uses of PhotoMesa (by JNLP tracking): **24,200**

## CounterPoint Presentation Tool

Most current commercial slide show presentation tools consist of a linearly ordered set of 'slides' that can be shown in sequence to an audience. There are also special mechanisms for moving back and forth in the sequence, jumping to a slide out of order (based on its title), and authoring hyperlinks in advance from any one slide to another.

Through the authors' experience, it has been found that zooming presentations naturally address several common problems with these presentation tools. These problems include: navigating to slides outside of a direct linear sequence during presentation delivery, maintaining multiple versions of very similar presentations, and differentiating levels of detail in presentation content. Although aware of workarounds in current tools to solve these problems, the authors believe that the zooming paradigm offers more elegant solutions.

Zoomable User Interfaces address these problems in several ways. First, because it stores presentations in a single contiguous space, a ZUI can help make jumping out of a linear presentation sequence possible through animated spatial navigation. Second, ZUIs support multiple presentation versions by allowing multiple paths through a single zoomable space. These multiple paths are possible since navigation is not directly tied to the presentation content. Third, ZUIs intrinsically provide for differentiated levels of detail by allowing information to be displayed at varying zoom levels.

The authors have created a tool called CounterPoint to simplify the authoring, management, and delivery of ZUI presentations. CounterPoint has facilities for hierarchically organizing presentation content to help automate spatial arrangement and assist in visually distinguishing levels of detail. CounterPoint also offers techniques for creating and managing paths through a populated presentation space. Lastly, CounterPoint augments standard controls for delivering a linear presentation by providing simplified navigations to support improvisation. Through use of CounterPoint and ZUIs for presentations the authors have found that these tools offer tremendous freedom for creativity. However, as with any creative medium, this flexibility introduces the possibility for bad design. Consequently, the authors have also begun to identify principles, from related work such as concept maps and their own experience, for the design of ZUI presentations.

<http://www.cs.umd.edu/hcil/counterpoint>

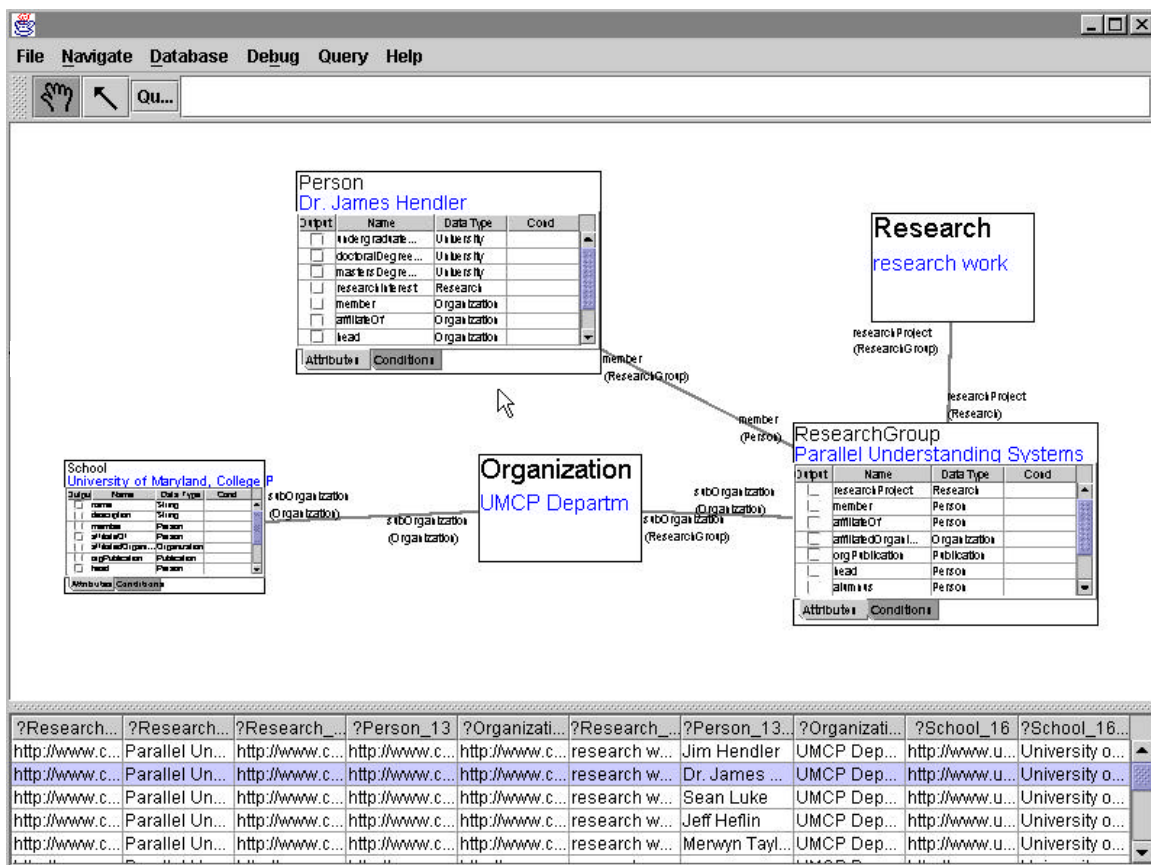
- Unique accessors total: **6,900**

- Unique accessors weekly average: **130**
- Unique downloads total: **520**
- Unique downloads weekly average: **10**

## OZONE Ontology Navigator

Information on the World Wide Web (WWW) has expanded enormously during the last several years. Searching the Web by string matching and link analysis has been one of the most popular ways of finding information in the vast quantity of data. But since these approaches rely on syntactic information, the search result of such schemes is often limited and ineffective. The Web has been designed for direct human processing. It is humans that write most of the web pages. Therefore, for accurate knowledge extraction, it is crucial to identify embedded semantic knowledge in them.

To address these issues, we built OZONE (Zoomable Ontology Navigator), a visual interface for the exploration of semantic information that is defined in DAML. OZONE reads ontology information from the Web and rearranges it visually with its context information so that ontology information can be queried and browsed easily and effectively. OZONE is implemented in pure Java using the Parka knowledgebase, the Jazz zoomable interface toolkit, and the API for XML Processing (JAXP) XML parser from Sun Microsystems.

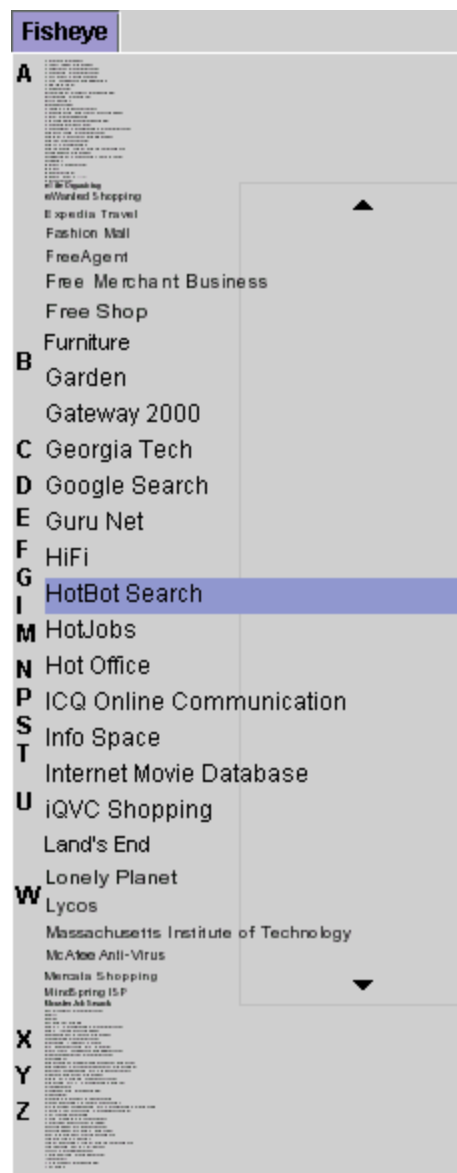


**OZONE: Zoomable Ontology Navigator**

## Fisheye Menus

We created “fisheye menus” which apply traditional fisheye graphical visualization techniques to linear menus. This provides for an efficient mechanism to select items from long menus, which are becoming more common as menus are used to select data items in, for example, e-commerce applications. Fisheye menus dynamically change the size of menu items to provide a focus area around the mouse pointer. This makes it possible to present the entire menu on a single screen without requiring buttons, scrollbars, or hierarchies.

A pilot study with 10 users compared user preference of fisheye menus with traditional pull-down menus that use scrolling arrows, scrollbars, and hierarchies. Users preferred the fisheye menus for browsing tasks, and hierarchical menus for goal-directed tasks.



Fisheye Menu for selecting items from long lists

<http://www.cs.umd.edu/hcil/fisheyemenu>

- Unique accessors total: **22,700**
- Unique applet viewers: **7,800**

## Published Papers

1. Hornbæk, K., Bederson, B. B., & Plaisant, C. (2002) Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview, *ACM Transactions on Computer-Human Interaction*, 9 (4), 362-389.
2. Bederson, B. B., Shneiderman, B., & Wattenberg, M. (2002) Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies, *ACM Transactions on Graphics*, 21 (4), 833-854, ACM Press.
3. Good, L., & Bederson, B.B. (2002) Zoomable User Interfaces as a Medium for Slide Show Presentations, *Information Visualization*, Palgrave Macmillan, pp. 35-49.
4. Suh, B., & Bederson, B. B. (2002) OZONE: A Zoomable Interface for Navigating Ontology Information. *Proceedings of International Conference on Advanced Visual Interfaces (AVI 2002)*, Trento, Italy, 139-143, ACM Press.
5. Bederson, B. B. (2001). PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. *UIST 2001, ACM Symposium on User Interface Software and Technology, CHI Letters*, 3(2), pp. 71-80.
6. Bederson, B. B., Meyer, J., & Good, L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters*, 2(2), pp. 171-180.
7. Bederson, B. B. (2000). Fisheye Menus. *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters*, 2(2), pp. 217-225.
8. Bederson, B. B., & Boltman, A. (1999). Does Animation Help Users Build Mental Maps of Spatial Information? *In Proceedings of Information Visualization Symposium (InfoVis 99)* New York: IEEE, pp. 28-35.
9. Combs, T. T. A., & Bederson, B. B. (1999). Does Zooming Improve Image Browsing? *In Proceedings of Digital Library (DL 99)* New York: ACM, pp. 130-137.

## Invited Presentations

1. September, 2002 **Microsoft Research Limited**, Cambridge, England  
"Information Visualization for Mobile Devices"

- 
2. July, 2002 **Microsoft Research**, Redmond, WA  
"Scalable Interfaces for Mobile Devices"
  3. June, 2002, **WebShop**, Dept. of Sociology, UMD, College Park, MD  
"Access and Usability: The Importance of Internet Applications"
  4. January, 2002 **Workshop on Election Standards and Technology**, Washington, DC  
"Visualization Techniques for Electronic Voting"
  5. October, 2001 **Microsoft Research**, Redmond, WA  
"Zoomable User Interfaces for Searching and Browsing: SearchKids & PhotoMesa"
  6. October, 2001 **Xerox PARC**, Palo Alto, CA  
"Zoomable User Interfaces for Searching and Browsing: SearchKids & PhotoMesa"
  7. November, 2000 **Census Bureau, Dept. of Commerce**, Suitland, MD  
"Zoomable User Interfaces"
  8. May, 2000 **Bowie State University**, Bowie, MD  
"Zoomable User Interfaces and Single Display Groupware"
  9. April, 2000 **University of Delaware**, DE  
"Zoomable User Interfaces and Single Display Groupware"
  10. August, 1999 **Xerox PARC**, Palo Alto, CA  
"Zoomable User Interfaces and Single Display Groupware"
  11. August, 1999 **University of California, Berkeley**, Berkeley, CA  
"Zoomable User Interfaces and Single Display Groupware"
  12. August, 1999 **Sun Microsystems, Inc. (Java Group)**, Palo Alto, CA  
"Zoomable User Interfaces and Single Display Groupware"
  13. July, 1999 **Nokia**, Linköping Sweden  
"Zoomable Information Retrieval Systems and Single Display Groupware"